Smoothing Measure with Lipschitz Constant in a Quadratic Augmented Lagrangian Algorithm

Lennin Mallma Ramirez*

November, 2025

Abstract

In this note we propose a condition, a smoothing measure with the Lipschitz constant (SMeLC), and a safeguard for the penalty parameter in a quadratic augmented Lagrangian algorithm (QALA) with safeguarded Lagrange multipliers. Our aim is to avoid ill-conditioning in the subproblems generated by QALA. Under differentiability and nonconvexity assumptions, we address equality constraints problems. Finally, we report computational experiments in which our approach successfully solves a class of benchmark problems on which several existing algorithms fail.

Key words: Nonlinear optimization, augmented Lagrangian, equality constraint, quadratic penalty, numerical experiments.

AMS subject classifications: 90C30, 65K05, 90C26.

1 Introduction and Preliminaries

In this work, we are interested in solving the nonlinear programming problem with equality constraints as follows:

$$\text{Minimize } f(x) \text{ subject to } c(x) = 0, \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ and $c\colon \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable functions. Problem (1) is solved, in particular by quadratic augmented Lagrangian algorithms. These algorithms are classical in the constrained optimization literature, see, for example, [14], [20] and [3]. The basic idea of these methods is to penalize the constraints, thus generating an unconstrained optimization problem in the primal variables and updating the Lagrange multipliers (dual variables). The Karush-Kuhn-Tucker (KKT) conditions for problem (1) are satisfied at a primal-dual pair $(x, \mu) \in \mathbb{R}^n \times \mathbb{R}^p$ if the following conditions are satisfied:

$$\nabla L(x, \mu) = 0, \tag{2}$$

$$c(x) = 0, \tag{3}$$

where the Lagrangian function is given by $L(x, \mu) = f(x) + \mu^T c(x)$ and $\mu \in \mathbb{R}^p$ are the Lagrange multipliers associated with x. The quadratic augmented Lagrangian function associated with problem (1), is defined as follows:

$$\mathcal{L}(x, \mu, \rho) = f(x) + \frac{\rho}{2} \sum_{i=1}^{p} \left[c_i(x) + \frac{\mu_i}{\rho} \right]^2, \tag{4}$$

*Systems Engineering and Computer Science Program (PESC)/COPPE, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil. email: lenninmr@gmail.com

where $\rho > 0$, is the penalty parameter. Its gradient is given by

$$\nabla_x \mathcal{L}(x, \mu, \rho) = \nabla f(x) + \nabla c(x) [\mu + \rho c(x)] = \nabla f(x) + \mu \nabla c(x) = \nabla L(x, \mu).$$

The QALA algorithms have already been thoroughly studied theoretically and computationally, and the safeguards applied to this algorithm make them even more competitive (see [3]). However, preventing the penalty parameter from growing indefinitely is a problem still under investigation. In [21], the authors say "of when it is possible to guarantee such a bound, that is, under what conditions the sequence of penalty parameters remains bounded..." (see, Page 16 of 43, Section 4 of [21]). We will briefly review this question and attempt to bound the penalty parameter. To this end, we will consider two types of mechanisms, as follows:

Controlled penalty mechanism

- Mukai and Polak [19] evaluate the decrease of the augmented Lagrangian function or the feasibility, to update the penalty parameter, thereby ensuring convergence for sufficiently large values of this parameter.
- Glad and Polak [11] use a test function (see pag. 141) and equations (16) and (17), control the growth of the penalty parameter.
- Sahin et al. [22] treat the penalty parameter as a regularization term, in their work, the penalty parameter and subproblem tolerance are synchronized (see Step 1 and Step 2 of Algorithm 1).

Unbounded growth of the penalty parameter ($\lim_{k\to\infty} \rho^k = +\infty$) or a "sufficiently large" penalty parameter:

- Conn, Gould and Toint [8] propose conditions (see equations (5.35), (5.36) and Theorem 5.3) that ensure the penalty parameter does not converge to zero, thereby avoiding the ill-conditioning of the subproblem generated by the algorithm.
- ALGENCAN [3] type algorithms (and extensions) use a well-known and widely adopted technique based on testing the reduction of feasibility violation at the new point (primal) point to update the penalty parameter, as follows: Let $0 < \theta < 1$ and $\nu > 1$. If $k = 0$ or

$$\|c(x^k)\|_\infty \leq \theta \|c(x^{k-1})\|_\infty, \tag{5}$$

  define $\rho^{k+1} = \rho^k$. Otherwise, define $\rho^{k+1} = \nu \rho^k$. For example see, Theorem 7.2 in [3], equation (4) in [4], equation (9) in [5] Algorithm 1 in [18], equation (2.9) in [12], Algorithm 2 in [9], Algorithm 2.1 in [16], equation (3.2) in [17], Algorithm 1 in [6], see Step 4 in Algorithm 2 and Algorithm 3 in [21].

The contributions of our work are as follows:

- The SMeLC introduces a safeguard mechanism based on a Lipschitz-type smoothing condition that controls the local variation of the augmented Lagrangian function and bounds the penalty parameter. This mechanism prevents the unbounded growth of the sequence $\{\rho^k\}$, which is a limitation in the quadratic augmented Lagrangian methods [3].
- The computational experiments, shows that incorporating the SMeLC condition, improves robustness of the quadratic augmented Lagrangian algorithm, and minimize the number of inner iterations. Consequently our results improves on the computational performance reported by the ALGENCAN algorithm in [21].

Our work is organized as follows. In Section 2, we present the SMeLC condition, which establishes a relationship between the Lipschitz constant, the smoothness of the augmented Lagrangian, and the penalty parameter, and we also prove a convergence result. In Section 3, we report computational experiments, on a class of benchmark problems from the literature. In Section 4, we present our conclusions and in Section 5, we outline directions for future research.

Notation. If $\ell, u \in \mathbb{R}^n$, we denote by $[\ell, u]$ the box-constraint $\{x \in \mathbb{R}^n \mid \ell \leq x \leq u\}$. We use $\|\cdot\|$ and $\|\cdot\|_\infty$ to denote the Euclidean and infinity norms, respectively.

2 SMeLC and Safeguarded Penalty Parameter

In this section, we maintain the safeguards technique on Lagrange multipliers (just as it is in the Step 4 in Algorithm 4.1 of [3]), and introduce our condition in the QALA algorithm, and also propose a safeguard on the penalty parameter.

Algorithm 1: *SMeLC-QALA*

Step 0. (Initialization) Let $\mu_{min} < \mu_{max}$, $\gamma > 1$, $\rho_{max} > 0$, $Tol > 0$, $\bar{\mu}^1 \in [\mu_{min}, \mu_{max}]^p$, and $\rho^1 > 0$.

Step 1. (Solve subproblem) Find $x^k \in \mathbb{R}^n$ as an approximation solution of

$$\|\nabla \mathcal{L}(x^k, \bar{\mu}^k, \rho^k)\|_\infty \leq Tol. \tag{6}$$

Step 2. (Estimate multipliers) Compute

$$\mu_i^{k+1} = \bar{\mu}_i^k + \rho^k c_i(x^k), \quad i = 1, ..., p. \tag{7}$$

Step 3. (Update penalty parameter) If

$$\frac{\left|\mathcal{L}(x^k, \bar{\mu}^k, \rho^k) - \mathcal{L}(x^{k-1}, \bar{\mu}^k, \rho^k)\right|}{\|x^k - x^{k-1}\| + 10^{-12}} \leq \frac{1}{\rho^k}, \tag{8}$$

choose $\rho^{k+1} = \rho^k$. Otherwise, define

$$\rho^{k+1} = \min\left\{\max\left\{\gamma\rho^k, 5\right\}, \rho_{max}\right\}. \tag{9}$$

Step 4. (safeguarded multipliers) Compute $\bar{\mu}^{k+1} \in [\mu_{min},\ \mu_{max}]^p$.

Step 5. (Check convergence) If

$$\|c(x^k)\|_\infty \leq \text{adaptive\_Tol}, \tag{10}$$

$$\|x^k - x^{k-1}\|_\infty \leq \text{adaptive\_Tol}, \tag{11}$$

or when

$$\|c(x^k)\|_\infty < 10^{-5}, \tag{12}$$

where

$$\text{adaptive\_Tol} = \max\left(Tol, \frac{10^{-6}}{1 + k^{0.5}}\right).$$

Step 6. (Continue) Set $k \leftarrow k+1$ and go to Step 1.

In Step 0, we enter the input data. In Step 1, we solve the subproblem using the Trust Region Method algorithm ([7]). In Step 2, we update the Lagrange multipliers. In Step 3, at each iteration k, we update the penalty parameter $\{\rho^k\}$ according to the variation measure (8), that is, if the SMeLC condition holds, the parameter $\{\rho^k\}$ remain constant, otherwise, it is increased up to a maximum safeguarded value ρ_{max}. This yields a bounded penalty strategy that improves the theoretical and numerical stability of the algorithm, ensuring convergence to a KKT point without requiring $\lim_{k\to\infty} \rho^k = +\infty$. On the other hand, the double safeguards on the penalty parameter defined in (9) tell us the following:

- $\rho^{k+1} \geq 5$ the penalty parameter is prevented from being too small, which helps preserve the coerciveness of the augmented Lagrangian.
- $\rho^{k+1} \leq \rho_{max}$, the penalty parameter will prevent the unlimited growth of parameter ρ^k.

An expression similar to (9) can be seen in Section 6.1 of [1] and the equation (2.9) in [12]. Something similar can also be observed, for the case of a smoothing parameter, where that parameter is safeguarded, for more details see Algorithm 2 of [18]. In Step 4, Lagrange multipliers are safeguarded (see Step 4 of Algorithm 4.1 [3]). In Step 5, the stopping criterion is checked. The adaptive_Tol is based on the tolerance rule studied in the Section 4 of [5], see also Choice 1 and Choice 2 of [10]. The following result, which is similar to Proposition 2.3 of [2] ensures that the sequence generated by SMeLC-QALA converges to a KKT point.

Theorem 2.1 *Let $\{x^k\}$, $\{\bar{\mu}^k\}$ and $\{\rho^k\}$ be the sequences generated by the SMeLC–QALA algorithm. Assume that:*

1. *$f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable;*
2. *$\{x^k\}$ is bounded and ρ^k is bounded;*
3. *$\lim_{k\to\infty} \|\nabla_x L(x^k, \bar{\mu}^k, \rho^k)\| = 0$;*
4. *$\lim_{k\to\infty} c(x^k) = 0$.*

Then, there exists a subsequence $\{k_j\}$ and a multiplier vector $\mu^* \in \mathbb{R}^p$ such that:

$$\lim_{j\to\infty} x^{k_j} = x^*, \quad \lim_{j\to\infty} \bar{\mu}^{k_j} = \mu^*, \quad \text{where } (x^*, \mu^*) \text{ satisfies the KKT conditions:}$$

$$\nabla f(x^*) + \nabla c(x^*)^\top \mu^* = 0, \quad c(x^*) = 0.$$

Proof: From Step 1 of the algorithm and (7), we have that for each k:

$$\nabla_x L(x^k, \bar{\mu}^k, \rho^k) = \nabla f(x^k) + \nabla c(x^k)(\bar{\mu}^k + \rho^k c(x^k)) = 0.$$

By assumption, $\|\nabla_x L(x^k, \bar{\mu}^k, \rho^k)\| \to 0$. Hence

$$\lim_{k\to\infty} \|\nabla f(x^k) + \nabla c(x^k)\mu^{k+1}\| = 0. \tag{13}$$

Because $\{x^k\}$ and $\{\bar{\mu}^k\}$ are bounded, there exists a subsequence $\{k_j\}$ such that $x^{k_j} \to x^*$ and $\bar{\mu}^{k_j} \to \mu^*$. Since ρ^k is bounded and $c(x^k) \to 0$, we have $\rho^k c(x^k) \to 0$. Hence

$$\mu^{k+1} = \bar{\mu}^k + \rho^k c(x^k) \to \mu^*.$$

Taking limits in (13) along $\{k_j\}$ we obtain

$$\nabla f(x^*) + \nabla c(x^*)\mu^* = 0.$$

From assumption, $c(x^*) = 0$. Therefore, (x^*, μ^*) satisfies the KKT condition. □

3 Computational Experiments

We implemented the SMeLC-QALA in Python. All experiments were carried out on a PC running Windows 11 with an 11th Gen Intel (R) Core (TM) i5-1135G7 @ 2.40GHz,
(2419 Mhz) processor and 16.0 GB, using compiler version 10.0.26100. We apply our code the Hock and Schittkowski problems (HS) test collection [15] with equality constraints. The subproblems generated by our algorithm were solved the Trust-Region method [7] from trust-ncg of SciPy. Gradient and Hessians were computed with autograd.

We used the following input data: $\rho^1 = 10$, $\gamma = 1.5$, $\rho_{max} = 10^6$ (an upper bounded that prevents the problem from becoming ill-conditioned), $\mu_{\min} = -10^6$, $\mu_{\max} = 10^6$, Tol $= 10^{-8}$ is the stopping criterion tolerance. The box-constraint $[-500, 500]$ was imposed to prevents numerical divergence of the internal solver (trust-region method). We also set max_ext_iters = 100 as the maximum number of external iterations and max_int_iters = 80 as the maximum number of internal iterations. Th initial point x^0 provided in [15] and $\mu^0 = 0$, in this section we present the results obtained by our algorithm, and compare them with the results reported in [13] and [21].

3.1 Equality constrained Hock and Schittkowski Problems (HS problems)

Let us consider the following notations.

- ALGENCAN: value reported by ALGENCAN in Table 1 of [21].
- AL 2: value reported by Algorithm 2 in Table 2 of [21].
- AL 3: value reported by Algorithm 3 in Table 3 of [21].
- ANRV: problem for which the authors do not report value.
- ADSWITCH: objective function value obtained by ADSWITCH in Table A.1 of [13].
- internal iters: number of internal interactions
- external iters: number of external iterations.

In Table 1, we report the values obtained by our algorithm. From Table 2, we observe that the average number of internal iterations is 20.7 for **SMeLC-QALA**, 56.5 for **ALGENCAN** (excluding ARNV), 64.2 for **AL 2** and 96.2 for **AL 3**. From Table 3, we observe that the average number of external iterations is 3.0 for **SMeLC-QALA**, 10.0 for **ALGENCAN**, 19.9 for **AL 2**, and 23.0 **AL 3**. From Table 4, we that our method attains values comparable to those reported in [15] and by the other algorithms. We also note that our proposal is more robust, as it successfully solves all problems, whereas the other algorithms either fail to convergence or do not report a solution for some instances; see, for example, HS46, HS49, HS56, and HS79. Moreover, Table 4 shows that the SMeLC condition preserves stability and yields consistent values, thereby highlighting the stabilizing effect of the SMeLC condition together with the bounded penalty parameter ρ.

4 Conclusions

In classical augmented Lagrangian methods (Hestenes, Powell, Rockafellar), convergence to a KKT point requires that the penalty parameter converge to infinity. However, this growth can cause numerical instability and/or ill-conditioned problems. Our computational experiments show that

Problem	$f(x^*)$	internal iters	external iters	Time (seg.)
HS06	2.750461e-12	70	1	1.179616e-01
HS07	-1.732051e+00	21	3	6.075668e-02
HS08	-1.000000e+00	8	1	1.675534e-02
HS09	-5.000000e-01	5	2	2.029133e-02
HS26	3.415727e-07	26	1	8.853817e-02
HS27	3.999968e-02	14	2	3.540587e-02
HS28	1.097010e-30	5	1	1.972580e-02
HS39	-1.000007e+00	33	7	1.349645e-01
HS40	-2.500014e-01	16	5	1.039071e-01
HS42	1.385786e+01	15	6	8.367205e-02
HS46	1.005830e-07	21	1	1.300361e-01
HS47	7.772446e-10	26	1	2.130337e-01
HS48	4.930381e-32	7	1	4.690480e-02
HS49	2.076266e-07	19	1	8.834553e-02
HS50	8.317497e-20	14	1	9.605265e-02
HS51	0.000000e+00	8	1	3.875875e-02
HS52	5.326627e+00	25	8	1.601672e-01
HS56	-3.455995e+00	29	4	2.460883e-01
HS61	-1.436461e+02	31	4	1.240151e-01
HS77	2.415050e-01	31	3	2.161529e-01
HS78	-2.919700e+00	18	4	1.766200e-01
HS79	7.877668e-02	14	3	1.178436e-01

Table 1: Numerical results of SMeLC-QALA on the all the HS equality constrained problems. All problems were solved in **2.37 seg.**

the SMeLC condition solves all tested problems with fewer external and internal iterations on average than techniques used in other quadratically augmented Lagrangian algorithms. This motivates further computational studies on large-scale problems, as well as an investigation of suitable initial parameters for SMeLC. Apply the SMeLC-QALA algorithm to solve large-scale problems. In this way, our work can be viewed as an improvement over the techniques employed in the ALGENCAN algorithm for the equality-constrained optimization problems

5 Future Works

Exploring the SMeLC condition in quadratic penalty algorithms under nondifferentiability assumptions, see for example algorithms in Part III of [23]. Adapting and incorporate the SMeLC condition into other classes of differentiable optimization algorithms.

Conflict of interest statement: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Funding: This work has been supported by FAPERJ (grant E-26/205.684/2022).

References

[1] Paul Armand and Riadh Omheni. A globally and quadratically convergent primal–dual augmented lagrangian algorithm for equality constrained optimization. *Optimization Methods and Software*, 32(1):1–21, 2017.

Problem	**SMeLC-QALA**	**ALGENCAN**	**AL 2**	**AL 3**
HS06	70	23	48	**110**
HS07	21	**49**	18	46
HS08	8	9	9	**40**
HS09	5	8	4	**14**
HS26	26	37	**337**	**660**
HS27	14	**55**	27	47
HS28	5	**33**	6	22
HS39	33	**125**	22	36
HS40	16	**54**	11	18
HS42	15	**33**	14	22
HS46	21	ARNV	ARNV	ARNV
HS47	26	**68**	25	46
HS48	7	**22**	6	**22**
HS49	19	51	58	**112**
HS50	14	20	11	**63**
HS51	8	19	8	**40**
HS52	25	**42**	22	32
HS56	29	34	**298**	**237**
HS61	31	18	16	**37**
HS77	31	**54**	21	51
HS78	18	**38**	14	36
HS79	14	**396**	**375**	**330**

Table 2: Internal Iterations

Problem	**SMeLC-QALA**	**ALGENCAN**	**AL 2**	**AL 3**
HS06	1	6	6	**13**
HS07	3	**15**	6	8
HS08	1	3	4	**13**
HS09	2	4	3	**5**
HS26	1	5	**100**	**100**
HS27	2	**8**	7	6
HS28	1	**7**	5	**7**
HS39	7	**20**	9	9
HS40	5	**8**	6	6
HS42	6	7	7	7
HS46	1	ARNV	ARNV	ARNV
HS47	1	7	6	**10**
HS48	1	6	4	**7**
HS49	1	7	**16**	24
HS50	1	5	4	**19**
HS51	1	6	6	**13**
HS52	8	8	**10**	**10**
HS56	4	**15**	**100**	**100**
HS61	4	6	6	**8**
HS77	3	**10**	7	8
HS78	4	8	6	**11**
HS79	3	**50**	**100**	**100**

Table 3: External Iterations

[2] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, Belmont, MA, 1996.

[3] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[4] E. G. Birgin and J. M. Martínez. Complexity and performance of an augmented Lagrangian algorithm. *Optimization Methods and Software*, 35(5):885–920, 2020.

[5] EG Birgin, Gabriel Haeser, and José Mário Martínez. Safeguarded augmented lagrangian algorithms with scaled stopping criterion for the subproblems. *Computational Optimization and Applications*, 91(2):491–509, 2025.

[6] Guido Cocchi and Matteo Lapucci. An augmented lagrangian algorithm for multi-objective optimization. *Computational Optimization and Applications*, 77(1):29–56, 2020.

[7] Thomas F Coleman and Yuying Li. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical programming*, 67(1):189–224, 1994.

[8] Andrew R Conn, Nicholas IM Gould, and Philippe Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[9] Andrea Cristofari, Gianni Di Pillo, Giampaolo Liuzzi, and Stefano Lucidi. An augmented lagrangian method exploiting an active-set strategy and second-order information. *Journal of Optimization Theory and Applications*, 193(1):300–323, 2022.

[10] Stanley C Eisenstat and Homer F Walker. Choosing the forcing terms in an inexact newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.

[11] Torkel Glad and Elijah Polak. A multiplier method with automatic limitation of penalty growth. *Mathematical Programming*, 17(1):140–155, 1979.

[12] Geovani Nunes Grapiglia and Ya-xiang Yuan. On the complexity of an augmented lagrangian method for nonconvex optimization. *IMA Journal of Numerical Analysis*, 41(2):1546–1568, 2021.

[13] Serge Gratton and Philippe L Toint. A simple first-order algorithm for full-rank equality constrained optimization. *arXiv preprint arXiv:2510.16390*, 2025.

[14] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[15] Willi Hock and Klaus Schittkowski. *Test examples for nonlinear programming codes*. Springer, 1987.

[16] Alexey F Izmailov, Mikhail V Solodov, and EI Uskov. Global convergence of augmented lagrangian methods applied to optimization problems with degenerate constraints, including problems with complementarity constraints. *SIAM Journal on Optimization*, 22(4):1579–1606, 2012.

[17] Christian Kanzow, Andreas B Raharja, and Alexandra Schwartz. An augmented lagrangian method for cardinality-constrained optimization problems. *Journal of Optimization Theory and Applications*, 189(3):793–813, 2021.

[18] Changshuo Liu and Nicolas Boumal. Simple algorithms for optimization on riemannian manifolds with constraints. *Applied Mathematics & Optimization*, 82(3):949–981, 2020.

[19] H Mukai and Elijah Polak. A quadratically convergent primal-dual algorithm with global convergence properties for solving optimization problems with equality constraints. *Mathematical Programming*, 9(1):336–349, 1975.

[20] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, UK, 1969.

[21] José Luis Romero, Damián Fernández, and Germán Ariel Torres. Enhancing sharp augmented lagrangian methods with smoothing techniques for nonlinear programming. *Journal of Optimization Theory and Applications*, 208(1):23, 2026.

[22] Mehmet Fatih Sahin, Ahmet Alacaoglu, Fabian Latorre, Volkan Cevher, et al. An inexact augmented lagrangian framework for nonconvex optimization with nonlinear constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Wim Stefanus van Ackooij and Welington Luis de Oliveira. Methods of nonsmooth optimization in stochastic programming. *International Series in Operations Research and Management Science*, 2025.

[24] Hao Wang, Fan Zhang, Jiashan Wang, and Yuyang Rong. An inexact first-order method for constrained nonlinear optimization. *Optimization Methods and Software*, 37(1):79–112, 2022.

Problem	**SMeLC-QALA**	**ALGENCAN**	**AL 2**	**AL 3**	**ADSWITCH**	[24]	Reference [15]
HS06	**2.750461E-12**	7.6980E-20	1.7749E-30	7.3651E-18	+9.238528E-13	1.503772E-13	0
HS07	-1.732051E+00	-1.7321E+00	-1.7321E+00	-1.7321E+00	-1.732051E+00	-1.732051E+00	-1.73205080757
HS08	-1.000000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.000000E+00	-1.000000E+00	-1
HS09	-5.000000E-01	-5.0000E-01	-5.0000E-01	-5.0000E-01	-5.000000E-01	-5.000000E-01	-0.5
HS26	3.415727E-07	3.6300E-13	1.0590E-08	1.5625E-08	+1.303916E-09	**8.505871E-06**	0
HS27	3.999968E-02	4.0000E-02	4.0000E-02	4.0000E-02	+3.999998E-02	4.000000E-02	0.04
HS28	1.097010E-30	1.5406E-18	2.3179E-24	2.0302E-18	**+9.687529E-13**	0.000000E+00	0
HS39	-1.000007E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.000000E+00	**-1.000044E+00**	-1
HS40	-2.500014E-01	-2.5000E-01	-2.5000E-01	-2.5000E-01	-2.500002E-01	-2.500000E-01	-0.25
HS42	1.385786E+01	1.3858E+01	1.3858E+01	1.3858E+01	+1.385786E+01	1.385786E+01	13.8578643763
HS46	1.005830E-07	**ARNV**	**ARNV**	**ARNV**	+8.808661E-09	**1.987922E-05**	0
HS47	7.772446E-10	8.7943E-19	2.2413E-20	4.1159E-19	+4.650485E-10	**2.842654E-05**	0
HS48	4.930381E-32	3.5793E-17	2.9251E-21	6.1576E-18	**+2.912966E-13**	1.109336E-31	0
HS49	2.076266E-07	5.4099E-10	6.2683E-13	1.4621E-12	**ARNV**	**4.978240E-03**	0
HS50	8.317497E-20	6.6509E-18	8.0396E-21	1.3296E-17	**+2.888537E-13**	1.232595E-32	0
HS51	0.000000E+00	2.8382E-20	5.6697-24	4.1029E-18	**+8.485715E-14**	2.170139E-08	0
HS52	5.326627E+00	5.3266E+00	5.3266E+00	5.3266E+00	+5.326648E+00	5.326649E+00	5.3266
HS56	-3.455995E+00	**1.3968E+47**	**1.0502E+00**	**-2.2250E+59**	**ARNV**	**-1.000000E+00**	-3.456
HS61	-1.436461E+02	-1.4365E+02	-1.4365E+02	-1.4365E+02	**ARNV**	-1.436462E+02	-143.6461422
HS77	2.415050E-01	2.4151E-01	2.4151E-01	2.4151E-01	+2.415051E-01	2.415043E-01	0.24150513
HS78	-2.919700E+00	-2.9197E+00	-2.9197E+00	-2.9197E+00	-2.919700E+00	-2.919700E+00	-2.91970041
HS79	7.877668E-02	**2.4224E+14**	**5.9982E+01**	**6.0361E+08**	+7.877683E-02	7.877686E-02	0.0787768209

Table 4: Objective function values